

The fixed-mesh ALE approach applied to solid mechanics and fluid–structure interaction problems

Joan Baiges^{*,†} and Ramon Codina

Technical University of Catalonia, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

SUMMARY

In this paper we propose a method to solve Solid Mechanics and fluid–structure interaction problems using always a fixed background mesh for the spatial discretization. The main feature of the method is that it properly accounts for the advection of information as the domain boundary evolves. To achieve this, we use an Arbitrary Lagrangian–Eulerian (ALE) framework, the distinctive characteristic being that at each time step results are projected onto a fixed, background mesh. For solid mechanics problems subject to large strains, the fixed-mesh (FM)-ALE method avoids the element stretching found in fully Lagrangian approaches. For FSI problems, FM-ALE allows for the use of a single background mesh to solve both the fluid and the structure. Copyright © 2009 John Wiley & Sons, Ltd.

Received 12 May 2009; Revised 22 July 2009; Accepted 22 July 2009

KEY WORDS: fixed-mesh methods; ALE formulations; immersed boundaries; fluid–structure interaction

1. INTRODUCTION

The fixed-mesh arbitrary Lagrangian–Eulerian (ALE) method (FM-ALE from now on) is a fixed grid method. Its main feature is that the domain movement is taken into account when computing the temporal derivatives. The basic idea consists in using an ALE strategy and remeshing at each time step in such a way that the original fixed mesh is recovered. This has two main advantages

*Correspondence to: Joan Baiges, Technical University of Catalonia, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.

†E-mail: jbaiges@cimne.upc.edu

Contract/grant sponsor: Ministerio de Ciencia e Innovación

Contract/grant sponsor: Col·legi d'Enginyers de Camins, Canals i Ports de Catalunya

when compared with other fixed grid methods:

1. As an ALE formulation is used, temporal derivatives can be correctly computed, including the convective terms arising due to the domain movement.
2. The values of the variables in previous time steps are clearly defined in the so-called *newly created nodes*, an issue of particular controversy in most fixed grid methods.

ALE formulations were initially developed for fluid dynamics problems, in which they were necessary to cope with fluid–structure interaction (FSI) and free surface problems (see [1–3]). In classical ALE methods, remeshing is often necessary after a certain number of time steps in order to avoid element stretching. The FM-ALE method avoids this need by projecting the results from the ALE deformed mesh onto a fixed background mesh *at each time step*, prior to solving the flow equations. At the end all the calculations can be performed on the fixed mesh, and in fact the ALE deformed mesh does not need to be explicitly built.

The FM-ALE method for flow problems in moving domains is extensively described in [4]: the main algorithmic steps of the method are set and two applications in the field of flow problems are presented. Here we take the same ideas and we apply them to problems in solid mechanics and FSI.

The most usual approach to solve solid mechanics problems is the use of Lagrangian formulations. This means that equations are written for *material* points following the movement of particles. This is a natural choice as in solid mechanics we are interested in tracking the behavior of structures in time (contrary to many problems of fluid mechanics where one is interested on the effect of the flow in a certain region, leading to Eulerian formulations). However, there are certain problems in which fully Lagrangian formulations cannot be used or lead to numerical difficulties: when a solid body is subject to large strains the shape of the elements that form the mesh can change a lot, resulting in stretched elements. Stretched elements cause that the system of equations to be solved is ill-conditioned, an inconvenience of particular importance if iterative methods are to be used. In this case ALE formulations are used and the mesh is no longer deformed following the particles, but is given an arbitrary movement that avoids the stretching of elements.

ALE methods for solid mechanics problems have been extensively developed (see [5–7]). The main concern in these works is to correctly compute the stress and plastic history variable update, as values of history variables at the previous time step are not available at the quadrature points unless a fully Lagrangian approach is used. In the framework of ALE strategies for solid mechanics problems, the FM-ALE method can be understood as an ALE method in which the mesh velocity is set to zero in all the domain except in the region close to the body surface. In [8] an Eulerian formulation for large deformation solid dynamics is presented. However, it is not clear how the issue of newly created nodes near the boundary is treated.

Once the FM-ALE strategy has been applied to both flow and solid mechanics problems, it is very natural to consider its use in the area of FSI. Several fixed grid strategies to solve FSI problems have been developed in the past years. As a first example, the immersed boundary method [9–13] consists in adding punctual penalty forces in the domain boundary so that the boundary conditions are fulfilled. Another possible approach is the use of Lagrange multipliers to enforce boundary conditions (see [14–16]). Both approaches are fictitious domain methods [17, 18] in the sense that the fluid–structure interface divides the fluid domain in a physical flow field and a fictitious field, which may be discretized and solved, but has no physical meaning to the FSI problem. Usually the unknown fields in this fictitious domain are used to assign values to the *newly created nodes* in the computation of time derivatives. In the *extended* finite element method, special functions are used to enrich the finite element space near the interface. In [19] a fixed mesh is used to solve

the fluid, while the solid is treated by a Lagrangian description. The description of the fluid–solid interface is done by means of a level set function. In all these works, a fully Lagrangian approach is used to deal with the solid.

An interesting feature of using the FM-ALE method to solve FSI problems is that as the regions occupied by the fluid and the solid do not superimpose, a single mesh can be used, giving some of the elements to the solid mechanics problem and the others to the flow problem. Special care has to be given to the coupling conditions between fluid and structure: the usual partitioned methods can be used with the FM-ALE method, although due to the fact that the same mesh is used to solve both problems a monolithic approach seems more suitable.

The paper is organized as follows. A review of ALE methods applied to solid mechanics problems is presented in Section 2. First the general ALE formulation is presented and particularized to the solid mechanics conservation laws. Afterwards the two possible approaches to face the equations are discussed: the monolithic approach deals with the arising equations in a classical manner, while in fractional-step methods the equations are solved in two steps, namely, the material and the convective phases. This allows for the use of specific numerical methods to solve each of the phases. In Section 3 the FM-ALE method is described. As a detailed explanation of the method can be found in [4], only the general algorithm and the particular features of its application to solid mechanics problems are presented. Stress is put on critical issues such as the imposition of boundary conditions or the tracking of the solid body surface. Section 4 deals with the FM-ALE method applied to FSI problems. The equations for the coupled problem are presented. A description of some of the most common coupling strategies and their particularization to FM-ALE follows. Finally in Section 5 some numerical examples and validation tests are carried out, showing the behavior of the proposed methodology. Some conclusions close the paper in Section 6.

2. ALE METHODS APPLIED TO SOLID MECHANICS

2.1. Problem statement

Let us consider a region $\Omega^0 \subset \mathbb{R}^d$ ($d=2,3$) where a solid body moves through during a time interval $[0, T]$. The solid at time t occupies only a subdomain $\Omega(t) \subset \Omega^0$. The boundary of $\Omega(t)$ is defined by a part of $\partial\Omega^0$ and a moving boundary that we call $\Gamma_{\text{free}} = \partial\Omega(t) \setminus \partial\Omega^0 \cap \partial\Omega(t)$.

In order to cope with the time-dependency of $\Omega(t)$, we use the ALE approach, with the particular feature of considering a variable definition of the domain velocity. Let χ_t be a family of invertible mappings, which for all $t \in [0, T]$ map a point $X \in \Omega(0)$ to a point $x = \chi_t(X) \in \Omega(t)$, with $\chi_0 = I$, the identity. If χ_t is given by the motion of the particles, the resulting formulation would be Lagrangian, whereas if $\chi_t = I$ for all t , $\Omega(t) = \Omega(0)$ and the formulation would be Eulerian.

Let now $t' \in [0, T]$, with $t' \leq t$, and consider the mapping

$$\begin{aligned}\chi_{t,t'} : \Omega(t') &\longrightarrow \Omega(t) \\ x' &\mapsto x = \chi_t \circ \chi_{t'}^{-1}(x')\end{aligned}$$

Given a function $f : \Omega(t) \times (0, T) \longrightarrow \mathbb{R}$, we define

$$\left. \frac{\partial f}{\partial t} \right|_{x'}(x, t) := \frac{\partial (f \circ \chi_{t,t'})}{\partial t}(x', t), \quad x \in \Omega(t), \quad x' \in \Omega(t')$$

In particular, the domain velocity taking as a reference the coordinates of $\Omega(t')$ is given by

$$\mathbf{u}_{\text{dom}} := \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\mathbf{x}'} (\mathbf{x}, t) \quad (1)$$

Three conservation laws are fundamental in solid mechanics, namely mass, momentum and energy balance. Let us make the assumption that mechanical effects are uncoupled from thermal effects. In this case, equations for mass and momentum balance can be solved independently from the energy balance equation. The solid mechanics problem formulated in $\Omega(t)$, accounting also for the motion of this domain, can be written as follows:

$$\left. \frac{\partial \rho}{\partial t} \right|_{\mathbf{x}'} + (\mathbf{u} - \mathbf{u}_{\text{dom}}) \cdot \nabla \rho = -\rho \nabla \cdot \mathbf{u} \quad (2)$$

$$\rho \left. \frac{\partial \mathbf{u}}{\partial t} \right|_{\mathbf{x}'} + \rho (\mathbf{u} - \mathbf{u}_{\text{dom}}) \cdot \nabla \mathbf{u} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (3)$$

where ρ is the solid density, \mathbf{u} is the particle velocity, $\boldsymbol{\sigma}$ is the Cauchy stress tensor and \mathbf{b} is the vector of body forces.

It is usual in the field of solid mechanics to use the following equation that relates the density ρ in a given configuration with the density ρ_0 at the undeformed configuration:

$$\rho J = \rho_0 \quad (4)$$

at each material point, where

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \quad J = \det(\mathbf{F})$$

As long as the material surfaces that compose the boundary of the solid are tracked with enough accuracy, this allows to avoid solving (2) and solving only for (3). An additional constitutive equation that relates $\boldsymbol{\sigma}$ and \mathbf{u} will be needed so that the problem is well posed.

If path dependent constitutive equations are to be used, material derivatives of the plastic internal variables have to account for the advection effects, leading to an equation for them of the form:

$$\left. \frac{\partial \boldsymbol{\alpha}}{\partial t} \right|_{\mathbf{X}} = \left. \frac{\partial \boldsymbol{\alpha}}{\partial t} \right|_{\mathbf{x}'} + (\mathbf{u} - \mathbf{u}_{\text{dom}}) \cdot \nabla \boldsymbol{\alpha} = \mathcal{F}(\boldsymbol{\lambda}) \quad (5)$$

where $\boldsymbol{\alpha}$ is the set of plastic internal variables and $\boldsymbol{\lambda}$ is the set of variables of interest of the problem, that would typically include the plastic internal variables plus the displacements, velocity and acceleration fields. The right-hand-side of (5) denotes a problem-dependent operator \mathcal{F} applied to $\boldsymbol{\lambda}$.

Initial and boundary conditions have to be appended to problem (3). Usual boundary conditions are used for both Γ_{free} and $\partial\Omega^0$:

$$\begin{aligned} \mathbf{u} &= \bar{\mathbf{u}} & \text{on } \Gamma_{\text{D}} \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \bar{\mathbf{t}} & \text{on } \Gamma_{\text{N}} \end{aligned} \quad (6)$$

where \mathbf{n} is the external normal to the boundary and $\bar{\mathbf{t}}$ are the given boundary data. Γ_{D} and Γ_{N} are, respectively, the Dirichlet and the Neumann parts of the boundary $\partial\Omega(t)$.

To shorten the notation, we will introduce the convection velocity

$$\mathbf{c} = \mathbf{u} - \mathbf{u}_{\text{dom}}$$

in what follows.

2.2. The time-discrete problem

Let us introduce some notation. Consider a uniform partition of $[0, T]$ into N time intervals of length δt . Let us denote by f^n the approximation of a time dependent function f at time level $t^n = n\delta t$. We also denote

$$\begin{aligned}\delta f^{n+1} &= f^{n+1} - f^n \\ \delta_t f^{n+1} &= \frac{f^{n+1} - f^n}{\delta t} \\ f^{n+\theta} &= \theta f^{n+1} + (1-\theta)f^n, \quad \theta \in [\tfrac{1}{2}, 1]\end{aligned}$$

θ type schemes: Suppose we are given a computational domain at time t^n , with spatial coordinates labeled \mathbf{x}^n , and an equation of the form:

$$\frac{\partial v}{\partial t} + \mathbf{c} \cdot \nabla v = \mathcal{G}(v)$$

where v is the unknown function and \mathcal{G} is an operator applied to it. If v^n is known, v^{n+1} can now be found as the solution of the problem:

Register for free at <https://www.scipedia.com> to download the version without the watermark

where now $\delta_t v^{n+1}|_{\mathbf{x}^n} = (v^{n+1}(\mathbf{x}) - v^n(\mathbf{x}^n))/\delta t$, being $\mathbf{x} = \chi_{t^{n+\theta}, t^n}(\mathbf{x}^n)$ the spatial coordinates in $\Omega(t^{n+\theta})$. The domain velocity given by (1), with $\mathbf{x}' = \mathbf{x}^n$, is approximated as

$$\mathbf{u}_{\text{dom}}^{n+\theta} = \frac{1}{\theta \delta t} (\chi_{t^{n+\theta}, t^n}(\mathbf{x}^n) - \mathbf{x}^n) \quad (8)$$

which allows us to compute $\mathbf{c}^{n+\theta} = \mathbf{u}^{n+\theta} - \mathbf{u}_{\text{dom}}^{n+\theta}$ in (7).

Fractional-step methods for solid mechanics: There are basically two ways of dealing with the ALE system of equations (2)–(6) (see [20] and the references therein):

- solving the fully coupled system of equations, accounting for the various terms simultaneously,
- using a fractional-step method to treat material and convective effects separately.

Although solving the coupled system of equations is more accurate, the fractional-step method offers some very useful advantages. On one hand, each of the equations to be solved is simpler than the ones arising from the coupled problem. On the other, difficulties on the computation of the stress field gradient, which are due to the fact that stresses are usually discontinuous across element edges, are more easily circumvented.

Remark 1

The FM-ALE method presented in this work has no dependence on the way the system of equations (2)–(6) is dealt with. However, for ease of implementation, fractional-step schemes have been chosen in the numerical examples presented in Section 5.

Let us consider the θ type scheme in (7). For simplicity we will consider $\theta = 1$. This equation can be solved in a monolithic way, but it can also be divided in two phases:

Material phase (first-order splitting): In the first phase we solve:

$$\frac{v^{n+1}(\mathbf{x}_{\text{mat}}) - v^n(\mathbf{x}^n)}{\delta t} = \mathcal{G}(v^{n+1}(\mathbf{x}_{\text{mat}})) \quad (9)$$

where $\mathbf{x}_{\text{mat}} = \mathbf{X}_{t^{n+1}, t^n}(\mathbf{x}^n)$ is the mapping given by the motion of the particles. Note that this first phase corresponds to $\mathbf{u}_{\text{dom}} = \mathbf{u}$ ($\mathbf{c} = \mathbf{0}$), that is to say, to a fully Lagrangian approach.

Convective phase (first-order splitting): In the second phase we solve:

$$\frac{v^{n+1}(\mathbf{x}) - v^{n+1}(\mathbf{x}_{\text{mat}})}{\delta t} + \mathbf{c}^{n+1}(\mathbf{x}) \cdot \nabla v^{n+1}(\mathbf{x}) = 0 \quad (10)$$

where $\mathbf{x} = \chi_{t^{n+1}, t^n}(\mathbf{x}^n)$ are the spatial coordinates in $\Omega(t^{n+1})$.

If we add (9) and (10) we obtain:

$$\frac{v^{n+1}(\mathbf{x}) - v^n(\mathbf{x}^n)}{\delta t} + \mathbf{c}^{n+1}(\mathbf{x}) \cdot \nabla v^{n+1}(\mathbf{x}) = \mathcal{G}(v^{n+1}(\mathbf{x}_{\text{mat}})) \quad (11)$$

which corresponds exactly to (7) except for the fact that instead of evaluating $\mathcal{G}(v^{n+1})$ at \mathbf{x} we evaluate it at \mathbf{x}_{mat} . This introduces an error of $\mathcal{O}(\delta t)$: observe from (10) that $\|v^{n+1}(\mathbf{x}) - v^{n+1}(\mathbf{x}_{\text{mat}})\| = \mathcal{O}(\delta t)$, where $\|\cdot\|$ may be taken, for example, as the L^2 -norm.

If one wants a second order in time scheme, this could be achieved by modifying scheme (9)–(10) in the following manner:

$$\frac{v^{n+1}(\mathbf{x}_{\text{mat}}) - v^n(\mathbf{x}^n)}{\delta t} + \mathbf{c}^n \cdot \nabla v^n(\mathbf{x}) = \mathcal{G}(v^{n+1}(\mathbf{x}_{\text{mat}})) \quad (12)$$

Convective phase (second-order splitting):

$$\frac{v^{n+1}(\mathbf{x}) - v^{n+1}(\mathbf{x}_{\text{mat}})}{\delta t} + \mathbf{c}^{n+1} \cdot \nabla v^{n+1}(\mathbf{x}) - \mathbf{c}^n \cdot \nabla v^n(\mathbf{x}) = 0$$

Note that $\mathbf{c}^{n+1} \cdot \nabla v^{n+1}(\mathbf{x}) - \mathbf{c}^n \cdot \nabla v^n(\mathbf{x})$ is expected to be, formally, of first order in δt , and therefore $\|v^{n+1}(\mathbf{x}) - v^{n+1}(\mathbf{x}_{\text{mat}})\| = \mathcal{O}(\delta t^2)$. Thus, when $v^{n+1}(\mathbf{x}_{\text{mat}})$ is used in (12) instead of $v^{n+1}(\mathbf{x})$, the resulting splitting error is expected to be $\mathcal{O}(\delta t^2)$. If an overall second-order scheme is to be used, $\theta = \frac{1}{2}$ must be chosen.

These fractional-step schemes can be introduced to the system of equations (2)–(6) and also for the plastic internal variables $\boldsymbol{\alpha}$ whose evolution equation is given by (5).

Newmark's method: If in the constitutive equation that relates the stress tensor $\boldsymbol{\sigma}$ with the set of variables of interest of the problem there is a dependence on the displacement field \mathbf{d} , that is to say, $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{d}, \boldsymbol{\alpha})$, (3) becomes a second order in time equation. In [21], Newmark presented a method to discretely approximate the velocity and acceleration (\mathbf{a}) at time t^{n+1} as a function of displacements (\mathbf{d}), velocity and acceleration at time t^n in a Lagrangian framework. These three

fields can be related in the continuous case by means of the equations

$$\left. \frac{\partial \mathbf{u}}{\partial t} \right|_X = \mathbf{a}, \quad \left. \frac{\partial \mathbf{d}}{\partial t} \right|_X = \mathbf{u}$$

Newmark method reads:

$$\begin{aligned} \mathbf{a}^{n+1} &= \frac{1}{\beta \delta t^2} [\mathbf{d}^{n+1} - \mathbf{d}^n - \mathbf{u}^n \delta t] - \left(\frac{1}{2\beta} - 1 \right) \mathbf{a}^n \\ \mathbf{u}^{n+1} &= \frac{\gamma}{\beta \delta t} [\mathbf{d}^{n+1} - \mathbf{d}^n] + \left(1 - \frac{1}{2\beta} \right) \delta t \mathbf{a}^n \end{aligned} \quad (13)$$

where β and γ are parameters to be chosen. Most usual values are $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$, which provide a second-order stable and non-dissipative scheme. In the case of displacement-dependent stress tensors, this method is to be used instead of θ type schemes. For the sake of conciseness, we will restrict what follows to θ schemes, both in the monolithic and fractional-step versions. The dependence of σ on the rest of variables of the problem (including internal variables α) will be simply indicated by $\sigma = \sigma(\lambda)$.

2.3. The fully discrete problem

The next step is to consider the spatial discretization of the time-discrete problem for both the coupled and fractional-step methods. Here we present the discretization obtained if finite elements are used.

Let $\{\Omega^e\}^{n+1}$ be a finite element partition of the domain $\Omega(t^{n+1})$, with index e ranging from 1 to the number of elements n_{el} . We denote with a subscript h the finite element approximation to the unknown functions. The test functions for the velocity \mathbf{u}_h will be denoted by \mathbf{v}_h , whereas γ_h will be the test functions for the discrete internal variables α_h , the finite element approximation to the solution of (5). All the unknowns and test functions are referred to the current configuration of the solid.

The standard Galerkin method applied to the monolithic time-discretized problem reads: find \mathbf{u}_h^{n+1} and α_h^{n+1} such that

$$m_s^{n+\theta}(\delta_t \mathbf{u}_h^{n+1} |_{x^n}, \mathbf{v}_h) + a_s^{n+\theta}(\lambda_h, \mathbf{v}_h) + c_s^{n+\theta}(\mathbf{c}_h; \mathbf{u}_h, \mathbf{v}_h) = l_s^{n+\theta}(\mathbf{v}_h) \quad (14)$$

$$(\delta_t \alpha_h^{n+1} |_{x^n}, \gamma_h) + (c_h^{n+\theta} \cdot \nabla \alpha_h^{n+\theta}, \gamma_h) = (\mathcal{F}^{n+\theta}(\lambda_h), \gamma_h) \quad (15)$$

for all appropriate test functions \mathbf{v}_h and γ_h . The different forms appearing in (14) are given by

$$m_s(\delta_t \mathbf{u}_h, \mathbf{v}_h) = \int_{\Omega} \mathbf{v}_h \cdot \rho \delta_t \mathbf{u}_h \, d\Omega$$

$$a_s(\lambda_h, \mathbf{v}_h) = \int_{\Omega} \nabla \mathbf{v}_h : \sigma(\lambda_h) \, d\Omega$$

$$c_s(\mathbf{c}_h; \mathbf{u}_h, \mathbf{v}_h) = \int_{\Omega} \mathbf{v}_h \cdot (\rho \mathbf{c}_h \cdot \nabla \mathbf{u}_h) \, d\Omega$$

$$l_s(\mathbf{v}_h) = \int_{\Gamma_N} \mathbf{v}_h \cdot \bar{\mathbf{t}} \, d\Omega + \int_{\Omega} \mathbf{v}_h \cdot \rho \mathbf{b} \, d\Omega$$

where \mathbf{c}_h is the discrete convection velocity, defined as:

$$\mathbf{c}_h = \mathbf{u}_h - \mathbf{u}_{\text{dom}}$$

The superscript $n + \theta$ in the different terms of (14) indicates the time level where unknowns and time-dependent functions need to be evaluated, as well as the spatial domain where integrals need to be performed. In (15) the symbol (\cdot, \cdot) denotes the L^2 -inner product in this spatial domain.

The test functions \mathbf{v}_h in (14) must vanish at the Dirichlet part of the boundary Γ_D . As \mathcal{F} in (15) is usually an algebraic operator, functions γ_h need to vanish only at points \mathbf{x}^n , at which the temporal derivatives in (14)–(15) are referred.

Remark 2

When diffusion is small in a convection–diffusion process or, as in the case of (15), the process is purely convective (which happens when \mathcal{F} is an algebraic operator), the Galerkin method fails and stabilized methods need to be used. The method we use is SUPG (see [22] for an overview of stabilization methods), which applied, for example, to (15) leads to the modification of this equation to

$$\begin{aligned} & (\delta_t \alpha_h^{n+1} |_{\mathbf{x}^n}, \gamma_h + \tau \mathbf{c}_h^{n+\theta} \cdot \nabla \gamma_h) + (\mathbf{c}_h^{n+\theta} \cdot \nabla \alpha_h^{n+\theta}, \gamma_h + \tau \mathbf{c}_h^{n+\theta} \cdot \nabla \gamma_h) \\ & = (\mathcal{F}^{n+\theta}(\lambda_h), \gamma_h + \tau \mathbf{c}_h^{n+\theta} \cdot \nabla \gamma_h) \end{aligned} \quad (16)$$

where the so-called stabilization parameter τ is computed elementwise as

$$\tau_e = \left(c \frac{|c|_e}{h_e} \right)^{-1}, \quad e = 1, \dots, n_{\text{el}}$$

where h_e is the element size for linear elements and half of it for quadratics and $|c|_e$ is a characteristic value of $|c|$ on element e . In the numerical experiments we have taken the algorithmic constant $c = 2$. Stabilization might also be necessary if the coupled method is used. However, it is not needed in most solid mechanics simulations as the convective term is usually not dominant in Equation (14). This method corresponds to the algebraic version of the subgrid scale approach (see [23]) and circumvents the stability problems of the Galerkin method. In particular, in the case of incompressible materials it is possible to use equal velocity pressure interpolations, that is, we are not tight to the satisfaction of the inf–sup stability condition (see Section 4.1).

For the fractional-step approach, Equations (14)–(15) may be split into material and convective phases. Using $\theta = 1$ and a first-order splitting, the former would consist in finding $\mathbf{u}_h^{L,n+1}$ and $\alpha_h^{L,n+1}$ such that

$$\frac{1}{\delta t} m_s^{L,n+1} (\mathbf{u}_h^{L,n+1} - \mathbf{u}_h^n, \mathbf{v}_h) + a_s^{L,n+1} (\lambda_h, \mathbf{v}_h) = l_s^{L,n+1}(\mathbf{v}_h) \quad \forall \mathbf{v}_h \quad (17)$$

$$\frac{1}{\delta t} (\alpha_h^{L,n+1} - \alpha_h^n, \gamma_h) = (\mathcal{F}^{L,n+1}(\lambda_h), \gamma_h) \quad \forall \gamma_h \quad (18)$$

where superscript L is used to denote that all variables, including domain integrals, are evaluated considering zero convection velocity. The convection step consists in finding \mathbf{u}_h^{n+1} and α_h^{n+1} such

that

$$\frac{1}{\delta t} m_s^{n+1} (\mathbf{u}_h^{n+1} - \mathbf{u}_h^{L,n+1}, \mathbf{v}_h) + c_s^{n+1} (\mathbf{c}_h; \mathbf{u}_h, \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \quad (19)$$

$$\frac{1}{\delta t} (\boldsymbol{\alpha}_h^{n+1} - \boldsymbol{\alpha}_h^{L,n+1}, \boldsymbol{\gamma}_h) + (\mathbf{c}_h^{L,n+1} \cdot \nabla \boldsymbol{\alpha}_h^{n+1}, \boldsymbol{\gamma}_h) = 0 \quad \forall \boldsymbol{\gamma}_h \quad (20)$$

In order to take the convective term linear, the convection velocity in this step may be taken as

$$\mathbf{c}_h^{L,n+1} = \mathbf{u}_h^{L,n+1} - \mathbf{u}_{\text{dom}}^{n+1}$$

Remark 3

Note that when \mathcal{F} is an algebraic operator, (18) is in fact an approximation to an ordinary differential equation, which corresponds to the time integration, usually at each numerical integration point, of the evolution equation for the internal variables. Obviously, options better than the simplest backward Euler scheme of (18) could be used. On the other hand, (20) simply represents the transport of the internal variables from the material configuration to the final configuration at t^{n+1} . There are models in which also the stresses $\boldsymbol{\sigma}_h$ need to be transported. As these stresses are discontinuous across the element edges for C^0 shape functions, solving Equation (16) for $\boldsymbol{\sigma}$ is not straightforward. There are a certain number of strategies to deal with this problem, which can be found, for example, in [20, 24, 25]. In the numerical examples of Section 5, there is no need to update the stresses, as only elastic materials have been considered. In this case, we compute the elastic strain from the displacement field, which is continuous for C^0 shape functions, and thus easily convected.

3. THE FM-ALE METHOD APPLIED TO SOLID MECHANICS

Register for free at <https://www.scipedia.com> to download the version without the watermark

An overview of the FM-ALE approach is presented, but major attention is given to the particular characteristics of its application to solids. For a more detailed explanation of the FM-ALE method in the general framework of moving domains, see [4].

In this section and the ones that follow it, the numerical schemes will be particularized for $\theta=1$.

3.1. The general algorithm

Suppose Ω^0 is meshed with a finite element mesh M^0 and that at time level t^n the domain $\Omega(t^n)$ is meshed with a finite element mesh M^n . Let \mathbf{u}_h^n be the velocity already computed on $\Omega(t^n)$. The purpose is to obtain the region the solid occupies at time t^{n+1} , $\Omega(t^{n+1})$, and to compute the various unknown fields. If the classical ALE method is used, M^n would deform to another mesh defined at t^{n+1} . In the FM-ALE approach, we do not use this mesh to compute the unknowns of the problem, but instead we re-mesh in such a way that the new mesh is, essentially, M^0 once again.

The steps of the algorithm to achieve the goal described are the following:

1. Define $\Gamma_{\text{free}}^{n+1}$ by updating the function that defines it.
2. Deform *virtually* the mesh M^n to M_{ALE}^{n+1} using the classical ALE concepts and compute the mesh velocity $\mathbf{u}_{\text{dom}}^{n+1}$.

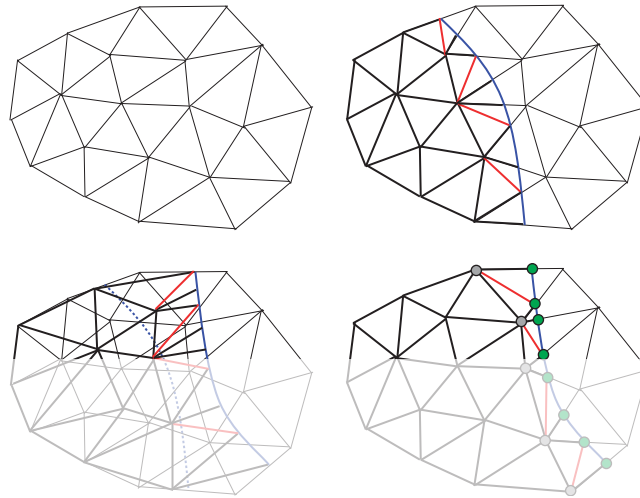


Figure 1. Two dimensional FM-ALE schematic. Top-left: original finite element mesh M^0 of Ω^0 . Top-right: finite element mesh M^n of $\Omega(t^n)$, with the elements represented by a thick line and the elements of M^0 represented by thin line. The blue line represents Γ_{free}^n and the red edges indicate the splitting of M^0 to obtain M^n . Bottom-left: updating of M^n to M_{ALE}^{n+1} using the classical ALE strategy. The position of $\Gamma_{\text{free}}^{n+1}$ is again shown using a solid blue line and the previous position Γ_{free}^n using a dotted blue line. Bottom-right: Mesh M^{n+1} of $\Omega(t^{n+1})$, represented by a thick line. The edges that split elements of M^0 are again indicated in red. Boundary nodes, where approximate boundary conditions need to be imposed, are drawn in green, whereas newly created nodes are drawn in gray.

Register for free at <https://www.solipedia.com> to download the version without the watermark

2. Write the FM-ALE solid mechanics equations on M_{ALE}^{n+1} .
4. Split the elements of M^0 cut by $\Gamma_{\text{free}}^{n+1}$ to define a mesh on $\Omega(t^{n+1})$, M^{n+1} .
5. Project the ALE solid mechanics equations from M_{ALE}^{n+1} to M^{n+1} .
6. Solve the equations on M^{n+1} to compute the unknowns.

A global idea of the meshes involved in the process is represented in Figure 1. Note in particular that at each time step two sets of nodes have to be appropriately dealt with, namely, the newly created nodes and the boundary nodes. Newly created nodes are those nodes that were exterior to the computational domain in the previous time step, but are inside the domain at the current time step, and thus they appear in the equations. Boundary nodes are the nodes resulting from the intersection of the boundary function with the elements of the mesh. Contrary to other fixed grid methods, newly created nodes are treated in a completely natural way using the FM-ALE approach: the value of the velocity there is directly given by the projection step from M_{ALE}^{n+1} to M^{n+1} . Boundary nodes require an appropriate imposition of boundary conditions.

An alternative algorithm that would lead to a very similar final result is:

1. Define $\Gamma_{\text{free}}^{n+1}$ by updating the function that defines it.
2. Deform the mesh M^n to M_{ALE}^{n+1} using the classical ALE concepts and compute the mesh velocity $\mathbf{u}_{\text{dom}}^{n+1}$.

3. Write down the ALE solid mechanics equations on M_{ALE}^{n+1} .
4. Solve the equations on M_{ALE}^{n+1} to compute the unknowns in the deformed mesh.
5. *Split the elements* of M^0 cut by $\Gamma_{\text{free}}^{n+1}$ to define a mesh on $\Omega(t^{n+1})$, M^{n+1} .
6. *Project* the results from M_{ALE}^{n+1} to M^{n+1} .

The conceptual idea of both algorithms is basically the same, the only difference being that in the first algorithm the equations are solved on M^{n+1} while in the second algorithm they are solved on M_{ALE}^{n+1} . However, the second approach is more convenient if non-linear systems of equations are to be solved. This is due to the fact that the projection from M_{ALE}^{n+1} to M^{n+1} is done only at the end of the time step, while in the first algorithm this projection has to be carried out at each iteration.

3.2. Details on some of the steps

3.2.1. Tracking of Γ_{free} . In the examples presented in Section 5, the body surface has been tracked by means of a Lagrangian boundary mesh. The intersection between the finite element mesh and the Lagrangian mesh is found at each time step. After the ALE solid equations have been solved, the Lagrangian contour mesh is deformed. The transmission of information between the two meshes is done by means of an L^2 projection. There are two possible ways of updating the position of the Lagrangian mesh nodes. The first approach consists in computing

$$\begin{aligned} \int_{\Gamma} \mathbf{v}_h \cdot \delta \mathbf{d}_L^{n:n+1} d\Gamma &= \int_{\Gamma} \mathbf{v}_h \cdot \delta \mathbf{d}_{\text{FE}}^{n:n+1} d\Omega \\ \mathbf{d}_L^{n+1} &= \mathbf{d}_L^n + \delta \mathbf{d}_L^{n:n+1} \end{aligned} \quad (21)$$

while in the second approach we compute

$$\int_{\Gamma} \mathbf{v}_h \cdot \mathbf{d}_L^{n+1} d\Gamma = \int_{\Gamma} \mathbf{v}_h \cdot \mathbf{d}_{\text{FE}}^{n+1} d\Omega \quad (22)$$

where \mathbf{d}_{FE} are the displacements computed on the finite element mesh and \mathbf{d}_L are the displacements of the Lagrangian surface mesh, $\delta \mathbf{d}^{n:n+1}$ are the incremental displacements from time step n to time step $n+1$ and \mathbf{v}_h are now the test functions corresponding to the nodes of the Lagrangian surface mesh.

Although (21) could seem a natural choice, as a usual approach in solid mechanics is to solve for the incremental nodal displacements, (22) works better than (21). This is a consequence of the fact that (22) preserves the information of the undeformed geometry, while the incremental approach of (21) leads to the loss of this information.

Another possible approach that has not been exploited in this work would be to track the body surface by means of a level set function. For more details on the use of level set functions in the FM-ALE method, see [26]. A method to track initial position of the particles has been developed in [27], which could also be applied to the present formulation.

3.2.2. Mesh velocity. Updating the boundary function defines the deformation of the domain from $\Omega(t^n)$ to $\Omega(t^{n+1})$. Consequently, the mesh M^n used at time step n has to be deformed to M_{ALE}^{n+1} .

The mesh velocity on the boundary points can be computed from their position \mathbf{x}_b^{n+1} and \mathbf{x}_b^n , where subscript b refers to points on Γ_{free} . The mesh velocity on these boundary points is

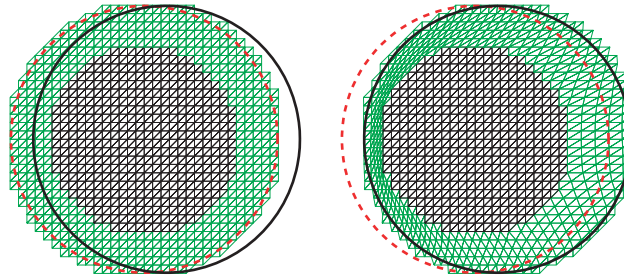


Figure 2. Mesh deformation. Left: M^n . Right: M_{ALE}^{n+1} . The dotted line represents the body surface at time t^n , the solid line corresponds to the body surface at time t^{n+1} —Black elements remain undeformed while the rest of elements are deformed from M^n to M_{ALE}^{n+1} .

$\mathbf{u}_{\text{dom},b}^{n+1} = (\mathbf{x}_b^{n+1} - \mathbf{x}_b^n) / \delta t$. Once the velocity at the nodes of Γ_{free} is known, it has to be extended to the rest of the nodes. A classical possibility is to solve the Laplace problem $\Delta \mathbf{u}_{\text{dom}} = 0$ using $\mathbf{u}_{\text{dom},b}^{n+1}$ as Dirichlet boundary conditions. However, it is also possible to restrict $\mathbf{u}_{\text{dom}} \neq \mathbf{0}$ to a region of Ω close to $\Gamma_{\text{free}}^{n+1}$, as in our approach mesh distortion does not accumulate from one time step to another. The condition we use in order to choose which of the nodes of the mesh are allowed to move is:

$$\text{dist}(\mathbf{x}, \Gamma_{\text{free}}) < K \cdot \max |\mathbf{u}_{\text{dom},b}| \cdot \delta t$$

where \mathbf{x} is the node position and $K > 1$ is a user-defined constant that adjusts the size of the region of Ω in which the mesh is deformed. This ensures that the mesh deformation is smooth enough for large values of $(\delta t/h) \max |\mathbf{u}_{\text{dom},b}|$, where h is the element size. An example of mesh deformation from M^n to M_{ALE}^{n+1} is represented in Figure 2.

Remark 4

As only nodes close to $\Gamma_{\text{free}}^{n+1}$ are displaced, the projection operations between meshes need only to be carried out in the deformed region of the mesh.

Remark 5

If large time steps are taken, it might be convenient to deform the full mesh at each time step. In this case, the projection operation needs to be carried out in the full $\Omega(t)$ region. When this happens one should choose:

$$\mathbf{u}_{\text{dom}} = \mathbf{u}_h \quad \text{and} \quad \mathbf{c} = \mathbf{0}$$

If fractional-step methods are used, this allows us to avoid the convective phase.

3.2.3. Splitting of elements. Mesh M^{n+1} is obtained by splitting the elements of M^0 cut by $\Gamma_{\text{free}}^{n+1}$. Meshes M^{n+1} and M^0 only differ in the subelements created after the splitting just mentioned. Mesh M^{n+1} could be thought as a local refinement of mesh M^0 to make it conform to the boundary $\Gamma_{\text{free}}^{n+1}$. As in other fixed grid methods, this computational complication can be avoided by prescribing boundary conditions on $\Gamma_{\text{free}}^{n+1}$ in an approximate way, although the local refinement

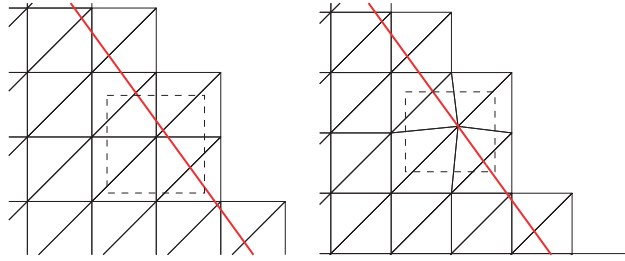


Figure 3. Deformation of M^0 for splitting purposes. Left: undeformed mesh. Right: deformed mesh. Element stretch is barely appreciable.

from M^0 to M^{n+1} is needed also to perform the numerical integration of the different terms appearing in (14)–(20).

However, depending on how $\Gamma_{\text{free}}^{n+1}$ intersects M^0 , the resulting subelements size could be very small compared with the size of elements adjacent to $\Gamma_{\text{free}}^{n+1}$. This results in an ill-conditioning of the system of equations to be solved. In order to avoid this issue, we work with a slightly deformed mesh $M_{\text{ALE,def}}^{n+1}$ at each time step constructed as follows: exterior nodes very close to $\Gamma_{\text{free}}^{n+1}$ (closer than $0.1h$ for example) are displaced in a direction orthogonal to $\Gamma_{\text{free}}^{n+1}$ until they match exactly the body surface. The splitting of this mesh will avoid ill-conditioned elements.

Remark 6

Note that as only nodes very close to $\Gamma_{\text{free}}^{n+1}$ are displaced, the stretch of the elements is negligible, as it can be seen in Figure 3.

3.2.4. Approximate imposition of boundary conditions. From the conceptual point of view, there is no problem in imposing exactly the boundary conditions on Γ_{free} . However, this requires the dynamic addition of mesh nodes (see Figure 1, where these nodes are drawn in green). It is very convenient from the implementation standpoint to avoid the explicit introduction of such nodes and to prescribe boundary conditions approximately. We summarize next a strategy to prescribe Dirichlet boundary conditions on a generic immersed boundary, that we denote by Γ . This strategy provides optimal order of accuracy and proves to be suitable for both flow and FSI problems. See [28] for more details.

Let u_h be the unknown solution of a problem posed in $\Omega \subset \Omega^0$ for which we want to prescribe a condition on Γ . Let Ω_Γ be the set of elements cut by Γ , which is split as $\Omega_\Gamma = \Omega_{\Gamma,\text{in}} \cup \Omega_{\Gamma,\text{out}}$, where $\Omega_{\Gamma,\text{in}} = \Omega \cap \Omega_\Gamma$ and $\Omega_{\Gamma,\text{out}}$ is the interior of $\Omega_\Gamma \setminus \Omega_{\Gamma,\text{in}}$. Note that $\Omega = \Omega_{\text{in}} \cup \Omega_{\Gamma,\text{in}}$. For simplicity, we will assume that the intersection of Γ with the element domains can be exactly represented by the classical isoparametric mapping. For the notation to be used, see Figure 4.

Suppose that the unknown u_h is interpolated as

$$\begin{aligned} u_h(\mathbf{x}) &= \sum_{a=1}^{n_{\text{in}}} I_{\text{in}}^a(\mathbf{x}) U_{\text{in}}^a + \sum_{b=1}^{n_{\text{out}}} I_{\text{out}}^b(\mathbf{x}) U_{\text{out}}^b \\ &= \mathbf{I}_{\text{in}}(\mathbf{x}) \mathbf{U}_{\text{in}} + \mathbf{I}_{\text{out}}(\mathbf{x}) \mathbf{U}_{\text{out}} \end{aligned}$$

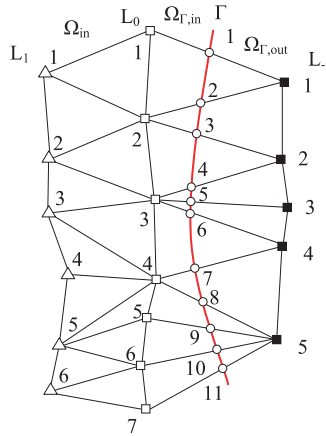


Figure 4. Immersed boundary sketch in a 2D example.

where $I_{\text{in}}^a(\mathbf{x})$ and $I_{\text{out}}^b(\mathbf{x})$ are the standard interpolation functions, n_{in} is the number of nodes in Ω_{in} , the domain where the problem needs to be solved (including layer L_0) and n_{out} the number of nodes in layer L_{-1} (see Figure 4).

The objective is to compute \mathbf{U}_{out} . Suppose that u_h needs to be prescribed to a given function \bar{u} on Γ . The main idea is to compute \mathbf{U}_{out} by minimizing the functional

$$J_2(\mathbf{U}_{\text{in}}, \mathbf{U}_{\text{out}}) = \int_{\Gamma} (u_h(\mathbf{x}) - \bar{u}(\mathbf{x}))^2 = \int_{\Gamma} (I_{\text{in}}(\mathbf{x})\mathbf{U}_{\text{in}} + I_{\text{out}}(\mathbf{x})\mathbf{U}_{\text{out}} - \bar{u}(\mathbf{x}))^2 \quad (23)$$

Suppose now that the problem for u_h in Ω_{in} leads to an algebraic equation of the form

$$\mathbf{K}_{\text{in},\text{in}}\mathbf{U}_{\text{in}} + \mathbf{K}_{\text{in},\text{out}}\mathbf{U}_{\text{out}} = \mathbf{F}_{\text{in}} \quad (24)$$

The domain integrals in matrices $\mathbf{K}_{\text{in},\text{in}}$ and $\mathbf{K}_{\text{in},\text{out}}$ extend only over Ω . The nodal values \mathbf{U}_{out} are merely used as degrees of freedom to interpolate u_h in the domain Ω . If (24) is supplemented with the equation resulting from the minimization of functional (23), the system to be solved is finally

$$\begin{bmatrix} \mathbf{K}_{\text{in},\text{in}} & \mathbf{K}_{\text{in},\text{out}} \\ \mathbf{N}_{\Gamma} & \mathbf{M}_{\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\text{in}} \\ \mathbf{U}_{\text{out}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_{\text{in}} \\ \mathbf{f}_{\Gamma} \end{bmatrix} \quad (25)$$

where

$$\mathbf{M}_{\Gamma} = \int_{\Gamma} \mathbf{I}_{\text{out}}^t(\mathbf{x}) \mathbf{I}_{\text{out}}(\mathbf{x}), \quad \mathbf{f}_{\Gamma} = \int_{\Gamma} \mathbf{I}_{\text{out}}^t(\mathbf{x}) \bar{u}(\mathbf{x}), \quad \mathbf{N}_{\Gamma} = \int_{\Gamma} \mathbf{I}_{\text{out}}^t(\mathbf{x}) \mathbf{I}_{\text{in}}(\mathbf{x})$$

It is important to note that this implementation maintains the connectivity of the background mesh.

As it is explained in detail in [28], this method works well if the boundary Γ is not too close to layer L_0 in Figure 4. In the case Γ is too close to L_0 , instabilities may appear in the solution field and remedies have to be devised. In this work, however, if a node is too close to layer L_0 it is moved in such a way that its final position is placed on Γ , as explained in Section 3.2.3. After this displacement the node no longer belongs to L_0 but to L_{-1} , and the instability problem disappears.

4. THE FM-ALE METHOD APPLIED TO FSI PROBLEMS

In [4] the FM-ALE method for solving flow problems in moving domains was presented. In this work we have seen how the FM-ALE approach can be used to solve problems in solid mechanics. In this section we will show how to solve FSI problems using the FM-ALE approach for both the fluid and the structure. In this case the same background fixed mesh can be used to solve both the first and the second problem, leading to some advantageous features in the coupling between them.

4.1. The FM-ALE method for flow problems in moving domains

In this section an incompressible Newtonian fluid will be considered. As in the solid case, the domain movement has to be taken into account. The incompressible Navier–Stokes equations are:

Find a velocity $\mathbf{u} : \Omega(t) \times (0, T) \longrightarrow \mathbb{R}^d$ and a pressure $p : \Omega(t) \times (0, T) \longrightarrow \mathbb{R}$ such that

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} \Big|_{\mathbf{x}'} + (\mathbf{u} - \mathbf{u}_{\text{dom}}) \cdot \nabla \mathbf{u} \right] - \nabla \cdot (2\mu \nabla^S \mathbf{u}) + \nabla p = \rho \mathbf{f} \quad (26)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (27)$$

where $\nabla^S \mathbf{u}$ is the symmetrical part of the velocity gradient, ρ is the fluid density, μ is the viscosity and \mathbf{f} is the vector of body forces. Initial and boundary conditions have to be appended to problem (26)–(27).

If finite elements are used, the fully discrete stabilized counterpart of these equations is (see [4] for details): find \mathbf{u}_h^{n+1} and p_h^{n+1} such that

$$m_1^{n+\theta}(\delta_t \mathbf{u}_h^{n+1} |_{\mathbf{x}^n}, \mathbf{v}_h) + a^{n+\theta}(\mathbf{u}_h, \mathbf{v}_h) + c^{n+\theta}(\mathbf{c}_h; \mathbf{u}_h, \mathbf{v}_h) + b_1^{n+\theta}(p_h, \mathbf{v}_h) = l_1^{n+\theta}(\mathbf{v}_h) \quad (28)$$

$$m_2^{n+\theta}(q_h, \delta_t \mathbf{u}_h^{n+1} |_{\mathbf{x}^n}) + b_2^{n+\theta}(q_h, \mathbf{u}_h) + s^{n+\theta}(q_h, p_h) = l_2^{n+\theta}(q_h) \quad (29)$$

for all test functions \mathbf{v}_h and q_h , the former vanishing on the Dirichlet part of the boundary Γ_D . The different forms appearing in these equations are given by

$$\begin{aligned} m_1(\delta_t \mathbf{u}_h, \mathbf{v}_h) &= \int_{\Omega} \mathbf{v}_h \cdot \rho \delta_t \mathbf{u}_h \, d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_{u1} \cdot \rho \delta_t \mathbf{u}_h \, d\Omega \\ a(\mathbf{u}_h, \mathbf{v}_h) &= \int_{\Omega} 2\nabla^S \mathbf{v}_h : \mu \nabla^S \mathbf{u}_h \, d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_{u1} \cdot (-2\nabla \cdot (\mu \nabla^S \mathbf{u}_h)) \, d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_{u2} \nabla \cdot \mathbf{u}_h \, d\Omega \\ c(\mathbf{c}_h; \mathbf{u}_h, \mathbf{v}_h) &= \int_{\Omega} \mathbf{v}_h \cdot (\rho \mathbf{c}_h \cdot \nabla \mathbf{u}_h) \, d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_{u1} \cdot (\rho \mathbf{c}_h \cdot \nabla \mathbf{u}_h) \, d\Omega \\ b_1(p_h, \mathbf{v}_h) &= - \int_{\Omega} p_h \nabla \cdot \mathbf{v}_h \, d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_{u1} \cdot \nabla p_h \, d\Omega \end{aligned}$$

$$\begin{aligned}
m_2(q_h, \delta_t \mathbf{u}_h) &= \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_p \cdot \rho \delta_t \mathbf{u}_h \, d\Omega \\
b_2(q_h, \mathbf{u}_h) &= \int_{\Omega} q_h \nabla \cdot \mathbf{u}_h \, d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_p \cdot (\rho \mathbf{c}_h \cdot \nabla \mathbf{u}_h - 2 \nabla \cdot (\mu \nabla^S \mathbf{u}_h)) \, d\Omega \\
s(q_h, p_h) &= \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_p \cdot \nabla p_h \, d\Omega \\
l_1(\mathbf{v}_h) &= \int_{\Omega} \mathbf{v}_h \cdot \rho \mathbf{f} \, d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_{u1} \cdot \rho \mathbf{f} \, d\Omega + \int_{\Gamma_N} \mathbf{v}_h \cdot \bar{\mathbf{t}} \\
l_2(q_h) &= \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \zeta_p \cdot \rho \mathbf{f}
\end{aligned}$$

where the functions ζ_{u1} , ζ_{u2} and ζ_p are computed within each element as

$$\zeta_{u1} = \tau_u [\rho \mathbf{c}_h \cdot \nabla \mathbf{v}_h + 2 \nabla \cdot (\mu \nabla^S \mathbf{v}_h)] \quad (30)$$

$$\zeta_{u2} = \tau_p \nabla \cdot \mathbf{v}_h \quad (31)$$

$$\zeta_p = \tau_u \nabla q_h \quad (32)$$

and the parameters τ_u and τ_p are also computed element-wise as (see [29])

$$\tau_u = \left[\frac{4\mu}{\rho h^2} + \frac{2|\mathbf{u}_h - \mathbf{u}_{\text{dom}}|}{h} \right]^{-1}, \quad \tau_p = 4\nu + 2|\mathbf{u}_h - \mathbf{u}_{\text{dom}}|/h, \quad \nu = \frac{\mu}{\rho}$$

where h is the element size for linear elements and half of it for quadratics.

4.2. Solving the coupled problem

When dealing with the coupled problem, the domain is divided into a solid part $\Omega_s(t)$ and a fluid part $\Omega_f(t)$, where $\bar{\Omega}^0 = \bar{\Omega}_s(t) \cup \bar{\Omega}_f(t)$ and $\Omega_s(t) \cap \Omega_f(t) = \emptyset$. The boundary of the coupled problem can now be divided into the Dirichlet boundary for the fluid Γ_D^f and for the solid Γ_D^s , the Neumann boundary for the fluid Γ_N^f and for the solid Γ_N^s , and the common interface boundary between the fluid and the solid Γ_{free} . The boundary of the coupled problem is now $\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_{\text{free}}$, where $\Gamma_D = \Gamma_D^f \cup \Gamma_D^s$ and $\Gamma_N = \Gamma_N^f \cup \Gamma_N^s$ (Figure 5).

The problem now consists in solving (14) or (17)–(20) in $\Omega_s(t)$ and (28)–(29) in $\Omega_f(t)$. The key point is obviously which are the boundary conditions to be applied. On Γ_D and Γ_N boundary conditions are usually applied to solid and fluid mechanics problems:

$$\begin{aligned}
\mathbf{u}^s &= \bar{\mathbf{u}}^s \quad \text{on } \Gamma_D^s \\
\mathbf{u}^f &= \bar{\mathbf{u}}^f \quad \text{on } \Gamma_D^f \\
\mathbf{n} \cdot \boldsymbol{\sigma}^s &= \bar{\mathbf{t}}^s \quad \text{on } \Gamma_N^s \\
\mathbf{n} \cdot \boldsymbol{\sigma}^f &= \bar{\mathbf{t}}^f \quad \text{on } \Gamma_N^f
\end{aligned}$$

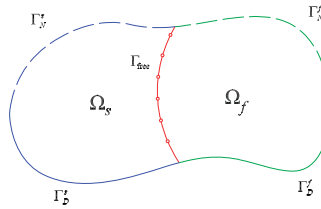


Figure 5. Domain and domain boundary subdivision in FSI problems.

where superscript s has been introduced for the unknowns in the solid and superscript f for the unknowns in the fluid. In Γ_{free} conditions must be applied such that velocity and traction continuity at all time steps is fulfilled:

$$\begin{aligned} \mathbf{u}^s &= \mathbf{u}^f \quad \text{on } \Gamma_{\text{free}} \\ \mathbf{n} \cdot \boldsymbol{\sigma}^s &= \mathbf{n} \cdot \boldsymbol{\sigma}^f \quad \text{on } \Gamma_{\text{free}} \end{aligned}$$

Satisfying the kinematic continuity leads to mass conservation, whereas satisfying the dynamic continuity yields conservation of linear momentum.

Note that as $\bar{\Omega}^0 = \bar{\Omega}_s(t) \cup \bar{\Omega}_f(t)$ and M^0 is a mesh covering Ω^0 , it is possible at each time step to divide M^0 into M_s^t and M_f^t such that:

$$M^0 = M_s^t \cup M_f^t$$

where M_s^t and M_f^t are meshes covering $\Omega_s(t)$ and $\Omega_f(t)$, respectively, and not necessarily disjoint. This allows us to use a single mesh M^0 to solve both the fluid and the solid mechanics problems for $t \in [0, T]$. However, if boundary conditions are prescribed in an approximate way, for example following the strategy proposed in Section 3.2.4, there will be some nodes of M^0 that will belong to both M_s^t and M_f^t . At these nodes (L_0 and L_{-1} nodes in Figure 4) degrees of freedom need to be duplicated so that unknowns for both the fluid and the solid can be obtained. A strategy to avoid the need of duplicating degrees of freedom is developed in [28], although it has not been used in the present work.

There are basically two ways of dealing with the coupled FSI problem: the *partitioned* and the *monolithic* approaches. In partitioned methods the solid and fluid problems are solved independently and coupling between both is achieved iteratively by means of the so-called *coupling algorithms*. The major advantage of this approach is that specific codes can be used for each of the two problems to be solved. Its drawback is that convergence is difficult to achieve under certain circumstances. In the monolithic approach, both problems are solved simultaneously and coupling between them is imposed in an *implicit* manner, which avoids the need of coupling iterations. The dimension of the system to be solved is larger in the monolithic case. However, if iterations within each time step yield convergence of the partitioned solution to the monolithic one, the distinction between both is blurred. In fact, those iterations can be understood as a certain preconditioner to solve iteratively the monolithic problem.

Although both strategies can be used together with the FM-ALE method, the monolithic approach is the one that suits it best. We have already seen how M^0 can be divided into M_s^t and M_f^t . Moreover, with the formulation we use to solve the incompressible Navier–Stokes equations, it

is possible to use the same interpolation functions for the unknowns corresponding to the solid problem and for the ones corresponding to the flow problem. It is very easy in this case to implicitly write the coupling conditions between fluid and structure. To this purpose equations corresponding to fluid velocity unknowns in nodes belonging to $L_{-1}^f(L_0^s)$ are used to prescribe implicitly $\mathbf{u}^s = \mathbf{u}^f$. On the other hand, traction continuity is imposed in equations corresponding to solid velocity/displacement in L_0 and L_{-1} simply by adding the corresponding boundary terms to the momentum conservation equations.

The final system to be solved is: find $\mathbf{u}_h^{n+1,s}$, $\mathbf{u}_h^{n+1,f}$ and p_h^{n+1} such that

$$\begin{aligned} m_1^{n+\theta}(\delta_t \mathbf{u}_h^f, \mathbf{v}_h^f) + a^{n+\theta}(\mathbf{u}_h^f, \mathbf{v}_h^f) + c^{n+\theta}(\mathbf{u}_h^f - \mathbf{u}_{\text{dom}}; \mathbf{u}_h^f, \mathbf{v}_h^f) + b_1^{n+\theta}(p_h, \mathbf{v}_h^f) &= l_1^{n+\theta}(\mathbf{v}_h^f) \\ m_2^{n+\theta}(q_h, \delta_t \mathbf{u}_h^f) + b_2^{n+\theta}(q_h, \mathbf{u}_h^f) + s^{n+\theta}(q_h, p_h) &= l_2^{n+\theta}(q_h) \\ m_s^{n+\theta}(\delta_t \mathbf{u}_h^s, \mathbf{v}_h^s) + a_s^{n+\theta}(\lambda_h, \mathbf{v}_h^s) + c_s^{n+\theta}(\mathbf{u}_h^s - \mathbf{u}_{\text{dom}}; \mathbf{u}_h^s, \mathbf{v}_h^s) &= l_s^{n+\theta}(\mathbf{v}_h^s) \end{aligned} \quad (33)$$

for all test functions \mathbf{v}_h^f and \mathbf{v}_h^s vanishing on the Dirichlet part of the boundary Γ_D , and all test functions q_h . Obviously, integrals corresponding to forms defined on the fluid region are extended over $\Omega_f(t)$, whereas integrals corresponding to forms associated with the solid are extended over $\Omega_s(t)$.

Another point we want to stress is that if one wants to solve the FSI problem using a monolithic scheme, but the solid is to be solved using a fractional-step method, the strategy to follow is simply to solve the *material* phase of the solid coupled monolithically with the fluid problem. Once this phase is solved, variables of interest in $\Omega_s(t)$ can be transported in the *convective* phase (only for the solid mechanics problem).

In the FSI examples in Section 5, the monolithic approach has been used. However, there is no major drawback in using the FM-ALE approach altogether with the partitioned schemes.

Let us close this section summarizing the final algorithm for the FM-ALE method applied to FSI problems, which is:

1. Define $\Gamma_{\text{free}}^{n+1}$ by updating the function that defines it.
2. Deform the mesh M^n to M_{ALE}^{n+1} using the classical ALE concepts and compute the mesh velocity $\mathbf{u}_{\text{dom}}^{n+1}$.
3. Write down the ALE solid (M_s^f) and fluid (M_f^f) mechanics equations on M_{ALE}^{n+1} . If a fractional-step method is used for the solid equations, this corresponds to the material phase.
4. Solve the equations on M_{ALE}^{n+1} to compute the unknowns in the deformed mesh.
5. If a fractional-step method is used for the solid, solve the convective phase.
6. *Split the elements* of M^0 cut by $\Gamma_{\text{free}}^{n+1}$ to define a mesh on $\Omega(t^{n+1})$, M^{n+1} .
7. *Project* the results from M_{ALE}^{n+1} to M^{n+1} .

5. NUMERICAL EXAMPLES

In this section we present some numerical examples that illustrate the behavior of the methodology proposed in this work.

5.1. An example of FM-ALE applied to solid mechanics

In this example a cantilever subject to gravity forces will be simulated by means of the FM-ALE method. As a fractional-step method is used, the equations to be solved are (17)–(19) (with the additional terms coming from the stabilization). A Neo-Hookean material has been considered, which takes into account large strains. The constitutive equation of this material is (see [30])

$$\boldsymbol{\sigma} = \frac{1}{J} [\lambda_0 \ln J \mathbf{I} + \mu_0 (\mathbf{B} - \mathbf{I})] \quad (34)$$

where $\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^T$, λ_0 and μ_0 are material parameters and \mathbf{I} is the identity tensor.

The hold-all domain is the rectangle $B = [-1, 5] \times [0, 11]$. A background mesh of 3200 linear triangles has been used. The considered solid is a rectangular cantilever situated at $[0, 1] \times [0, 10]$. At each time step there will be mesh nodes outside the solid domain. These nodes are ‘deactivated’ nodes, and they do not appear in the equations: at each time step we solve only for nodal unknowns in interior nodes and nodes belonging to the first exterior layer of nodes.

The material parameters are $\lambda_0 = 2000$ and $\mu_0 = 5000$. The solid density is $\rho = 1$, which has been considered to remain constant through the whole process. In this case the body is only under the effect of (horizontal) body forces, given by $\mathbf{b}^T = (1, 0)$. Dirichlet boundary conditions are applied at $y = 0$, where displacements in any direction are prescribed to zero. On the rest of the solid boundary, Neumann boundary conditions are applied:

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{0}$$

Initial conditions correspond to the undeformed static configuration.

The time step size has been set to $\delta t = 0.2$ and $\theta = 1$ has been taken (first order scheme in time). Figure 6 shows the mesh used to solve this problem. The boundary of the body does not match the boundary of the mesh. Figure 7 shows horizontal and vertical displacements at time step 90.

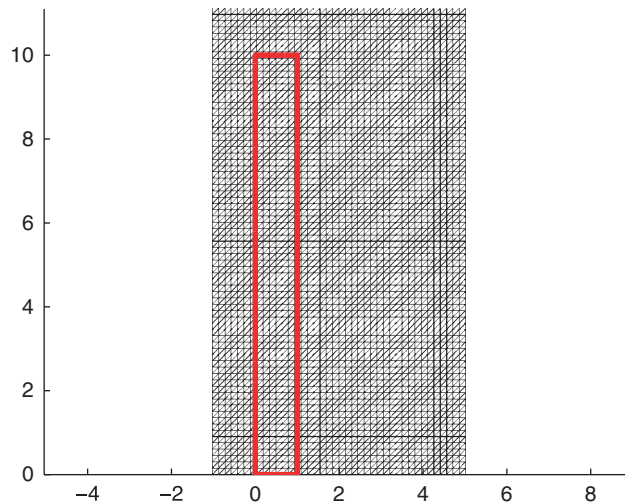


Figure 6. Immersed mesh use to solve the solid mechanics example and body surface.

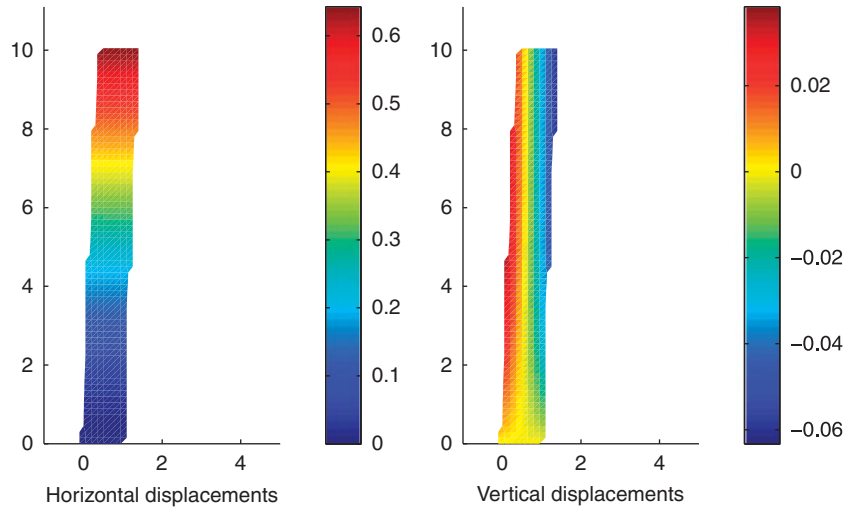


Figure 7. Displacements after 90 time steps.

In order to validate the FM-ALE method, we have compared the results obtained from our approach with those obtained if a classical Updated-Lagrangian method for boundary fitting meshes is used. To this end we have used a boundary fitting mesh with the same element density to solve the same problem. Figure 8 shows the horizontal displacement of a material point placed at the top of the cantilever, whose coordinates are $\mathbf{X}^T = (0.5, 10)$.

The element density in both meshes is relatively coarse. This is so in order to show that the fact that the mesh does not fit the domain does not introduce significative error in the solution. This is better seen in coarse meshes than in very fine meshes, where the boundary error would be less noticeable.

The simulation is carried out during 90 time steps. As it can be seen results are very similar to the ones obtained in the classical Updated-Lagrangian approach.

Another issue that we were interested in is the effect of the use of a fractional-step method. In Figure 9 we have plotted the results obtained if we take $\mathbf{u}_{\text{dom}} = \mathbf{u}_h$ (the convective phase is avoided) versus the results obtained if we take $\mathbf{u}_{\text{dom}} \neq \mathbf{u}_h$ (a convective phase is needed). As we can see, no difference can be appreciated between results, and we can conclude that the error introduced by the use of a fractional-step method is small.

5.2. Two examples of FM-ALE applied to FSI problems

In the first example the same cantilever as in Section 5.1 is simulated. However, the forces acting on the cantilever are due to the interaction with a fluid in this case. The hold-all domain is the rectangle $B = [-10, 70] \times [0, 20]$. An unstructured background mesh of 4655 linear triangles is used. This mesh is much coarser than the one used in the previous example, if we consider the element density in the solid body area.

The material parameters for the solid body are the ones used in the previous example. For the fluid we have considered $\rho = 2$ and $\mu = 0.2$. The velocity at $x = 0$ is prescribed to $(1, 0)$, whereas at $y = 0$ and $y = 20$ the y -velocity component is prescribed to 0 and the x -component is left free.

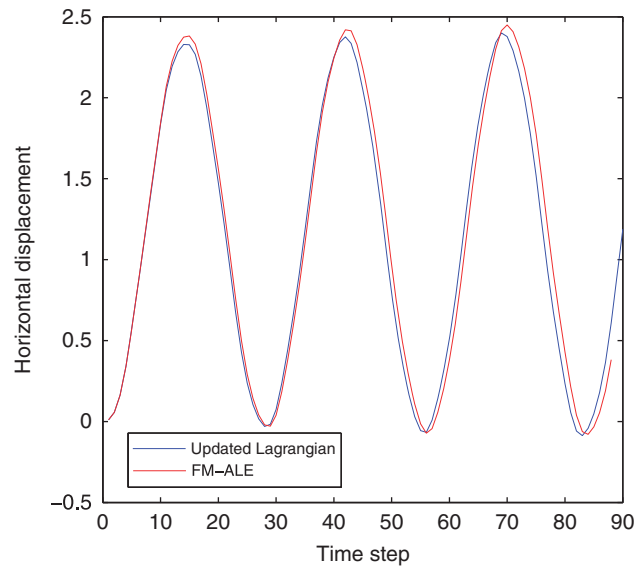


Figure 8. Horizontal displacement at a point placed at the top of the cantilever. Comparison between FM-ALE and Updated-Lagrangian formulations.

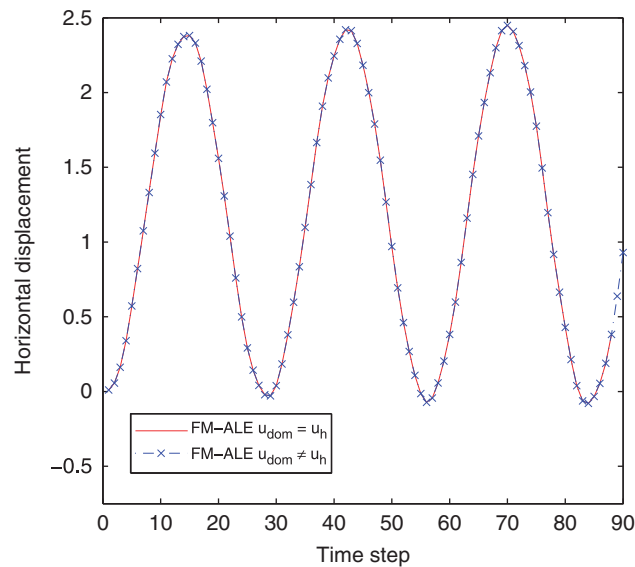


Figure 9. Horizontal displacement at a point placed at the top of the cantilever. Comparison between FM-ALE $\mathbf{u}_{\text{dom}} = \mathbf{u}_h$ and FM-ALE $\mathbf{u}_{\text{dom}} \neq \mathbf{u}_h$ formulations.

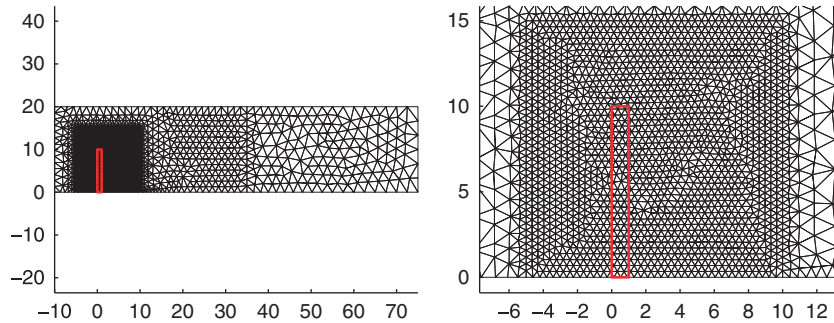


Figure 10. Mesh used to solve the first fluid–structure interaction example. Left: full mesh. Right: detail of the area surrounding the solid body.

The outflow (where both the x - and y -components are free) is $x=70$. The Reynolds number is 100, based on the cantilever height and the prescribed inflow velocity. The time step size has been set to $\delta t=1$ and $\theta=1$.

A monolithic approach has been used to couple fluid and structure. Both the fluid and the structure have been solved using the FM-ALE method with the *same background mesh* (see Figure 10). The deformed configuration of the beam at time step 100 is shown in Figure 11.

In Figure 12 the horizontal displacement at a point placed on the top of the cantilever is plotted. This figure shows how the movement of the cantilever is damped by the action of the fluid. After a certain number of time steps, the movement becomes stationary. Figures 13 and 14 show the cantilever displacements and the fluid velocities and pressures at time step 100. The irregular boundaries in Figure 13 are due to the fact that for ease of post processing, we have plotted the solution in the elements cut by the boundary without taking into account that the boundary of the domain does not fit the boundary of the elements.

In the second example we consider a thin elastic non-linear beam (Neo-Hookean material) attached to a fixed square rigid body, which is submerged in an incompressible fluid flow. Vortices separating from the corners of the rigid body generate oscillating forces on the beam. Geometry is given in Figure 15, while Figure 16 shows the mesh used to solve the problem. Again, a higher element density has been used in the region that will be occupied by the solid. The mesh is as coarse as possible, with the requirement that there are at least three elements to cover the beam width, and it is composed of 9388 triangular elements and 4812 nodes. The setting of the problem is similar to that proposed in [31], although we have considered a thicker beam in order to be able to use the rather coarse mesh described. It is clear that thin structures are not the most favorable situation for fixed-mesh methods, and in particular for FM-ALE (an alternative would be to represent these structures by a zero width solid, which is not a situation considered in this work).

The fluid material properties are $\rho=1 \times 10^{-2}$ and $\mu=1.7 \times 10^{-3}$. The solid material properties are $\rho=2$, $\lambda_0=1.72 \times 10^6$ and $\mu_0=7.4 \times 10^5$. The horizontal inflow velocity at $x=0$ is set to 40, yielding a Reynolds number of $Re=235$ referred to the length of the square rigid body. Slip boundary conditions are set at the walls of the channel. The beam and the square rigid body are assigned non-slip boundary conditions. The time step is set to $\delta t=0.002$ and $\theta=1$. A monolithic

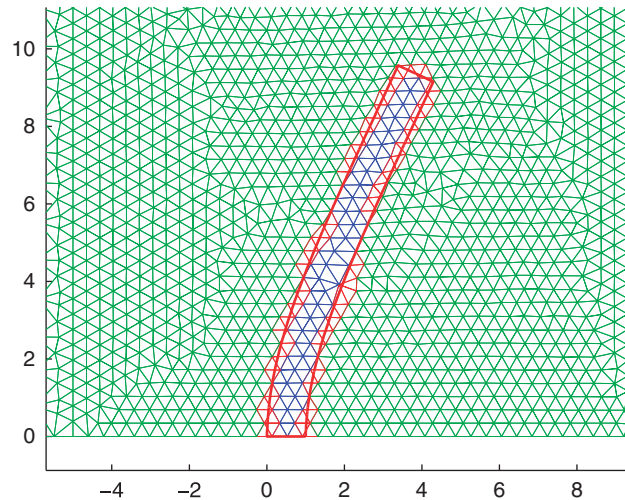


Figure 11. The same mesh is used to solve both the fluid and the structure. Elements in the fluid domain are used to obtain the fluid mechanics problem unknowns, while elements in the solid domain are used to obtain the solid mechanics problem unknowns. Elements in the interface need to be solved for both fluid and solid mechanics problems unknowns.

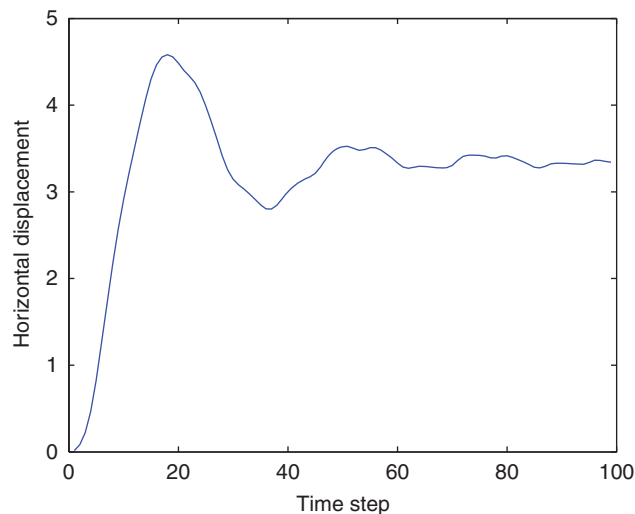
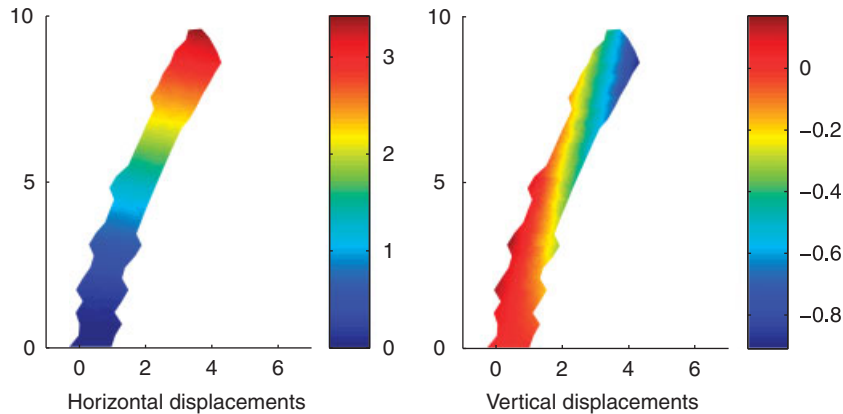
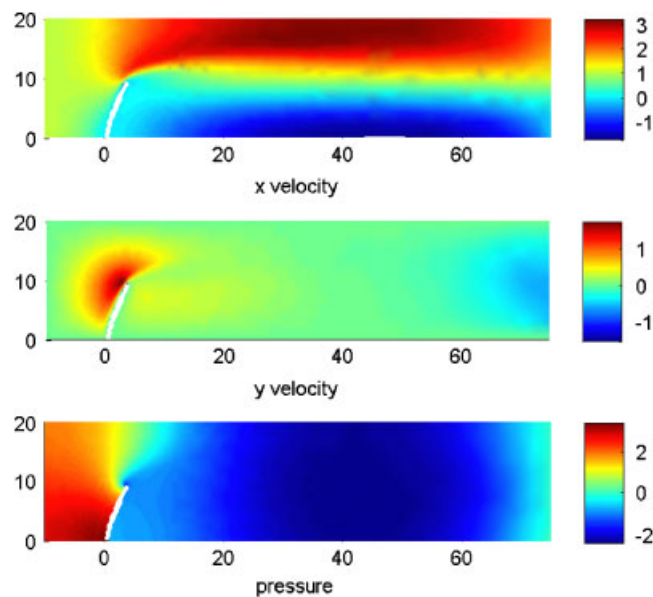


Figure 12. Horizontal displacement at a point placed at the top of the cantilever.

approach has been used to solve the FSI problem, although the fractional-step scheme has been used to deal with the solid. Both the fluid and the structure have been solved using the fixed background mesh.

Figure 13. Solution for the solid body at $t = 100$.Figure 14. Solution for the fluid at $t = 100$.

Contours of pressure and velocity components at $t = 5$ are shown in Figure 17, when the vortex shedding behind the square cylinder has appeared but is not yet fully developed. It can be observed that even in this transient stage results are smooth and boundary conditions on the elastic beam perfectly accounted for. The evolution of the vertical displacement at the edge of the beam is plotted in Figure 18, where it can be observed that the dynamics of the system are fully developed

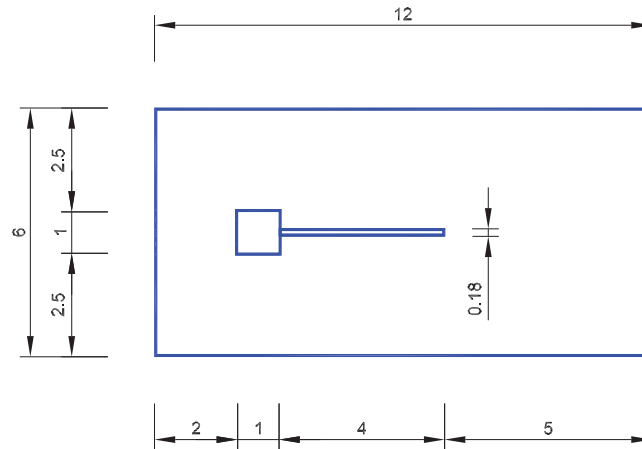


Figure 15. Geometry.

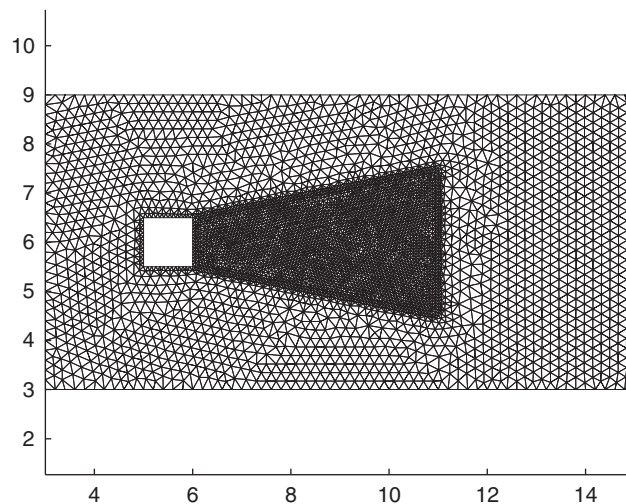


Figure 16. Mesh used to solve the second fluid–structure interaction example.

at about $t=7$. Then, a perfectly harmonic flow pattern sets in, with a single frequency in the time response, as it can be observed from Figure 19.

6. CONCLUSIONS

In this paper the FM-ALE approach has been applied to solid mechanics and FSI problems. The main feature of the method is its capability of using a fixed background mesh, but at the same time correctly taking into account the domain movement in the computation of the time derivatives.

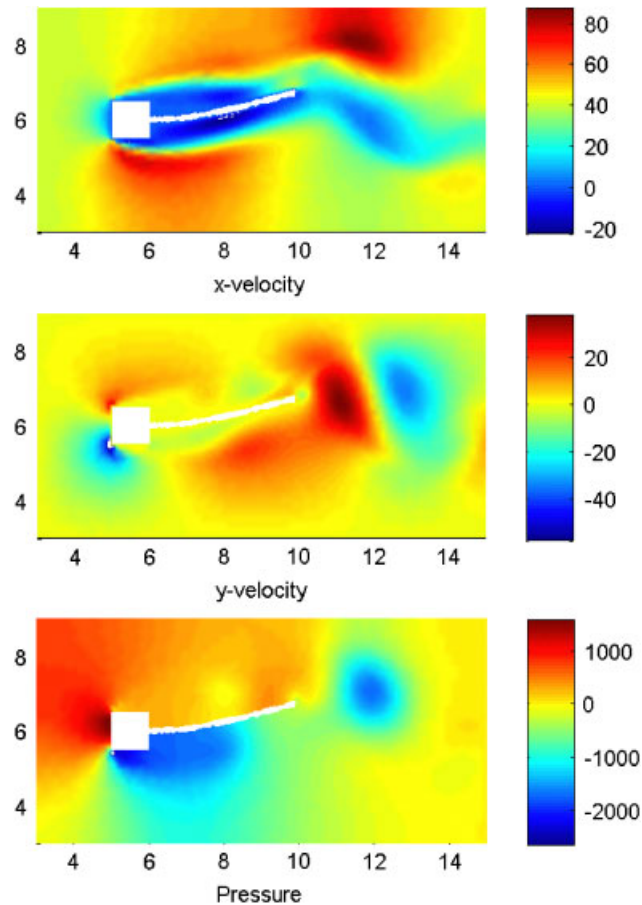


Figure 17. Velocity and pressure at time $t = 5.0$.

Moreover, values of the unknowns for the so-called *newly created nodes* are clearly and uniquely defined with the FM-ALE approach.

For solid mechanics problems, the FM-ALE method is of interest when the solid body is subject to very large strains. In this case Lagrangian formulations cannot be used due to the ill-conditioning caused by the large element stretch, and, if strains are large enough, even ALE formulations would require remeshing. The FM-ALE method, on the other hand, avoids element stretching by using a fixed mesh. A validation test has been carried out comparing results obtained with a classical Updated-Lagrangian formulation and the method proposed in this work. Results show that the method is robust and accurate.

The FM-ALE concept can be applied together with any time integration scheme. In the case of solid mechanics, we have shown how to use it in combination with classical θ schemes and fractional-step methods that have a certain popularity in this context.

In the case of FSI problems, the FM-ALE method can be applied to solve the flow and the solid mechanics problems. The main feature of this approach is the possibility of using a *single*

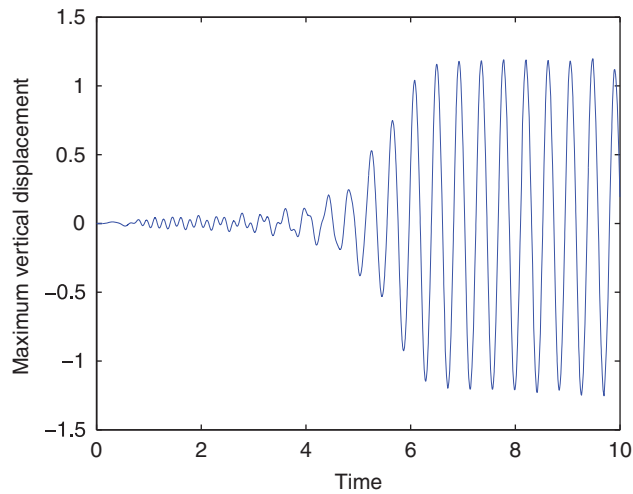


Figure 18. Vertical displacement at the edge of the beam.

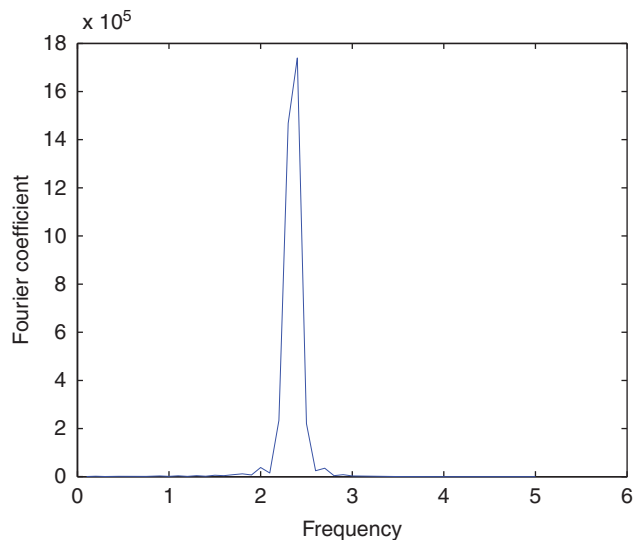


Figure 19. Fourier transform of the maximum vertical displacement. The frequency of the main vibration mode is 2.4.

background mesh to solve both mechanical problems. We have presented two numerical examples showing this particular capability. Even though monolithic solid–fluid coupling schemes have been employed, the possibility of using iteration-by-subdomain techniques is open.

ACKNOWLEDGEMENTS

J. Baiges would like to acknowledge the support received from the *Ministerio de Ciencia e Innovación* of the Spanish Government and the *Collegi d'Enginyers de Camins, Canals i Ports de Catalunya*.

REFERENCES

1. Donea J, Fasoli-Stella P, Giuliani S. Lagrangian and Eulerian finite element techniques for transient fluid structure interaction problems. *Transactions Fourth SMIRT*, San Francisco, 1977; B1/2.
2. Huerta A, Liu WK. Viscous flow with large free surface motion. *Computer Methods in Applied Mechanics and Engineering* 1988; **69**:277–324.
3. Hughes TJR, Liu WK, Zimmerman TK. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering* 1981; **29**:329–349.
4. Codina R, Houzeaux G, Coppola-Owen H, Baiges J. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics* 2009; **228**:1591–1611.
5. Liu WK, Belytschko T, Chang H. An arbitrary Lagrangian–Eulerian finite element method for path-dependent materials. *Computer Methods in Applied Mechanics and Engineering* 1986; **58**:227–245.
6. Benson DJ. An efficient, accurate, simple ALE method for nonlinear finite element programs. *Computer Methods in Applied Mechanics and Engineering* 1989; **72**:305–350.
7. Ghosh S, Kikuchi N. An arbitrary Lagrangian–Eulerian finite element method for large deformation analysis of elastic–viscoplastic solids. *Computer Methods in Applied Mechanics and Engineering* 1991; **86**:127–188.
8. Okazawa S, Kashiwayama K, Kaneko Y. Eulerian formulation using stabilized finite element method for large deformation solid dynamics. *International Journal for Numerical Methods in Engineering* 2007; **72**:1544–1559.
9. Peskin CS. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics* 1972; **10**:252–271.
10. Lai M-C, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics* 2000; **160**:705–719.
11. LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis* 1994; **31**(4):1019–1044.
12. LeVeque RJ, Li Z. Immersed interface method for incompressible Navier–Stokes equations. *SIAM Journal on Scientific and Statistical Computing* 1997; **18**(3):709–735.
13. Xu S, Wang ZJ. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics* 2006; **216**:454–493.
14. Dolbow J, Mourad HM, Harari I. A bubble-stabilized finite element method for Dirichlet constraints on embedded interfaces. *International Journal for Numerical Methods in Engineering* 2007; **69**:772–793.
15. Barbosa HJC, Hughes TJR. The finite element method with Lagrangian multipliers on the boundary: circumventing the Babuška–Brezzi condition. *Computer Methods in Applied Mechanics and Engineering* 1991; **85**:109–128.
16. Ji H, Dolbow JE. On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method. *International Journal for Numerical Methods in Engineering* 2004; **61**:2508–2535.
17. Glowinski R, Pan T-W, Périaux J. A fictitious domain method for Dirichlet problems and applications. *Computer Methods in Applied Mechanics and Engineering* 1994; **111**:203–303.
18. Glowinski R, Pan TW, Hesla TI, Joseph DD, Périaux J. A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application to particulate flow. *International Journal for Numerical Methods in Fluids* 1999; **30**:1043–1066.
19. Legay A, Chessa J, Belytschko T. An Eulerian–Lagrangian method for fluid–structure interaction based on level sets. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:2070–2087.
20. Rodríguez-Ferran A, Casadei F, Huerta A. ALE stress update for transient and quasi-static processes. *International Journal for Numerical Methods in Engineering* 1998; **43**:241–262.
21. Newmark NM. A method of computation for structural dynamics. *Journal of Engineering Mechanics Division (ASCE)* 1959; **85**:67–94.
22. Codina R. Comparison of some finite element methods for solving the diffusion–convection–reaction equation. *Computer Methods in Applied Mechanics and Engineering* 1998; **156**:185–210.
23. Hughes TJR. Multiscale phenomena: Green’s function, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**:387–401.

24. Huétink J. On the simulation of thermo-mechanical forming processes. *Dissertation*, 1986.
25. Armero F, Love E. An arbitrary Lagrangian–Eulerian finite element method for finite strain plasticity. *International Journal for Numerical Methods in Engineering* 2003; **57**:471–508.
26. Coppola-Owen H, Codina R. A finite element model for free surface flows on fixed meshes. *International Journal for Numerical Methods in Fluids* 2007; **54**:1151–1171.
27. Dunne T, Rannacher R. Adaptive finite element approximation of fluid–structure interaction based on an Eulerian variational formulation. In *Fluid–Structure Interaction: Modelling, Simulation, Optimisation*, Bungartz HJ, Schäfer J (eds). Lecture Notes in Computational Science and Engineering, vol. 53. Springer: Heidelberg, 2006.
28. Codina R, Baiges J. Approximate imposition of boundary conditions in immersed boundary methods. *International Journal for Numerical Methods in Engineering* 2009; DOI: 10.1002/nme.2662.
29. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:4295–4321.
30. Belytschko T, Liu WK, Moran B. *Nonlinear Finite Elements for Continua and Structures*. Wiley: New York, 2001.
31. Wall WA, Ramm E. Fluid–structure interaction based upon a stabilized (ALE) finite element method. In *Proceedings of the Fourth World Congress on Computational Mechanics (WCCM IV)*, Oñate E, Idelsohn SR (eds), Buenos Aires, Argentina, 29 June–2 July 1998. CIMNE: Barcelona, 1998; 1988–1995.